

MULTI-STANDARD VARIABLE BLOCK SIZE MOTION ESTIMATION PROCESSOR

This application claims the benefit of U.S. Provisional Application No. 60/487,643, filed July 15, 2003 and is hereby
5 incorporated by reference in its entirety.

Field of the Invention

The present invention relates to a digital video motion estimation generally and, more particularly, to a multi-standard
10 variable block size motion estimation method and apparatus.

Background of the Invention

Motion estimation is employed to reduce temporal redundancy in digital video compression and is therefore a central
15 part of the MPEG-x and the H.26x video coding standards. Conventional motion estimation computations command 60 to 80% of an encoding computational load. For large picture formats (i.e., standard definition and above) dedicated VLSI-architectures for motion estimation are employed. Numerous architectures exist for
20 fixed block-size motion estimation (FBSME), which is based on a

03-0838
1496.00329

constant, translational motion model with fixed-size image blocks, usually 8 horizontal (H) by 8 vertical (V) or 16Hx16V samples. A variable block size motion estimation (VBSME) processes, that allows a wider range of block sizes (i.e., 4Hx4V, 4Hx8V, 8Hx4V, 5 8Hx8V, 8Hx16V, 16Hx8V, and 16Hx16V), provides better estimation of small and irregular motion fields resulting in a reduced number of bits used for coding prediction errors compared with FBSME. The VBSME process is a common component of motion models in recent video compression standards, such as H.264 and MPEG-4 Part 2, yet 10 motion estimation processors that support VBSME are rare.

Two known VBSME approaches are currently available. A first VBSME approach uses a separate processing step for each block size. However, the first approach expends significant processing resources as a reference search area is traversed repeatedly, once 15 for each block-size. A second VBSME approach operates on a smallest block-size and derives all larger block-size sum of absolute differences (SAD) by summing the corresponding small block-size SADs. Since the intermediate results from the small block-sizes are kept to generate the full block-size SADs, the 20 second approach utilizes significant intermediate storage.

Summary of the Invention

The present invention concerns an apparatus generally comprising a first circuit, a second circuit and a third circuit. The first circuit may be configured to generate a plurality of
5 difference values by calculating an absolute difference between each pixel from a current block and a corresponding pixel from a reference block substantially simultaneously. The second circuit may be configured to generate a plurality of sum values by adding the difference values. The third circuit may be configured to
10 generate at least one motion vector in response to the sum values.

The objects, features and advantages of the present invention include providing a method and/or apparatus for variable block size motion estimation that may (i) calculate an array of absolute differences (e.g., 256 values for 16x16 pixels) for a
15 macroblock per cycle, (ii) calculate multiple sums of absolute differences for a smallest block size, (iii) calculate the absolute differences and the sum of absolute differences (SAD) in difference circuits, (iv) build larger block-size SADs from smaller block-size SADs, (v) provide lookup tables to bias the variable block size
20 SADS and the related motion vectors (e.g., to favor fewer, smaller motion vectors) for improved compression efficiency, (vi) reduce

03-0838
1496.00329

motion estimation process cycles, (vii) operate from a single pass through a reference search window, (viii) reduce an amount of temporary storage in the motion estimation processor as intermediate SAD values for partial block sizes may not be stored, (ix) input smaller block-size SADs into a larger block size SAD calculation logic without using any memory and/or (x) allow a motion estimation processor to support variable block size motion estimation with the addition of (number of blocks-1) comparators and registers.

10

Brief Description of the Drawings

These and other objects, features and advantages of the present invention will be apparent from the following detailed description and the appended claims and drawings in which:

15 FIG. 1 is a block diagram of multiple partition modes of a macroblock;

 FIG. 2 is a block diagram of an apparatus in accordance with a preferred embodiment of the present invention;

20 FIG. 3 is a block diagram of an example implementation of a processing element circuit;

FIG. 4 is a block diagram of an example implementation of an adder circuit; and

FIG. 5 is a block diagram of an example implementation of a mode decision tree circuit.

5

Detailed Description of the Preferred Embodiments

Referring to FIG. 1, a block diagram of multiple partition modes of a macroblock 100 is shown. A first partition mode (e.g., MODE0) generally defines the macroblock 100 as a single (e.g., 16Hx16V) block 102. Another partition mode (e.g., MODE1) may divide the macroblock 100 into two horizontal (e.g., 16Hx8V) sub-blocks 104a-104b. A partition mode (e.g., MODE2) generally divides the macroblock 100 into two vertical (e.g., 8Hx16V) sub-blocks 106a-106b. A fourth partition mode (e.g., MODE3) may divide the macroblock 100 into four square (e.g., 8Hx8V) sub-blocks 108-108d. A partition mode (e.g., MODE4) generally divides the macroblock 100 into eight horizontal (e.g., 8Hx4V) sub-blocks 110a-110h. Another partition mode (e.g., MODE5) may divide the macroblock 100 into eight vertical (e.g., 4Hx8V) sub-blocks 112a-112h. A partition mode (e.g., MODE6) may divide the

macroblock 100 into sixteen square (e.g., 4Hx4V) sub-blocks 114a-114p.

Referring to FIG. 2, a block diagram of an apparatus 120 is shown in accordance with a preferred embodiment of the present invention. The apparatus (or system) 120 generally comprises a
5 memory circuit (or module) 122, first circuit (or module) 124, a second circuit (or module) 126 and a third circuit (or module) 128. The apparatus 120 generally receives a signal (e.g., REF_PIXEL) carrying reference (or previous) blocks 130 of pixels at an input
10 132. The apparatus 120 may also receive a signal (e.g., ORIG_PIXEL) carrying current (or new) blocks 134 of pixels at an input 135. A signal (e.g., MV(x,y)) may be generated at an output 138 of the apparatus 120. A signal (e.g., MODE) may be generated at an output 140 of the apparatus 120. A signal (e.g., SAD) may be generated at an output 142 of the apparatus 120.

15 The signal REF_PIXEL may transfer reference frames that have been previously encoded and then decoded. The frames may represent one or more pictures used for motion estimation. Each frame generally comprises multiple reference blocks 130. Each reference block 130 may be implemented as a macroblock. The
20 reference blocks 130 may be arranged in a single pixel resolution, a half pixel resolution, a quarter pixel resolution or the like.

The signal ORIG_PIXEL may transfer current frames to be encoded. Each current frame generally comprises multiple current blocks 134. Each current block 134 may be implemented as a macroblock.

5 The signal MODE may define the particular partition mode resulting in the best motion estimation for the current block 134. The signal MODE may identify seven different modes (e.g., MODE0-MODE6) in designs of the apparatus 120 that may be capable of partitioning each of the current blocks 134 into sixteen smallest
10 sub-blocks (e.g., 114a-114p). The signal MODE may identify four different modes (e.g., MODE0-MODE3) in designs of the apparatus 120 that may be capable of partitioning each of the current blocks 134 into four 8x8 sub-blocks (e.g., 108a-108d). Other numbers of modes may be implemented to meet the criteria of a particular
15 application.

 The signal MV(x,y), or MV for short, may define one or more motion vectors for the current block 134 with respect to the reference frame or frames. The number of motion vectors within the signal MV may be defined by a particular partition mode determined
20 to be a best mode for the current block 134. For example, if the apparatus 120 determines that MODE0 should be used for motion

03-0838
1496.00329

estimation, the signal MV may convey a single motion vector for the current block 134. If the apparatus 120 determines that MODE3 should be used for motion estimation, the signal MV may convey four motion vectors, one for each of the four sub-blocks (e.g., 108a-
5 108d) within the current block 134.

The signal SAD may define a sum of absolute difference value for the best mode determined by the apparatus 120. The signal SAD may include a bias value and/or a cost value incorporated before determining the best mode. The bias value and
10 the cost value may be determined from quantization parameters and the motion vectors.

The memory circuit 122 may be configured as an internal reference memory to store the reference blocks 130. The memory circuit 122 may include the input 132 to receive the reference
15 blocks 130 via the signal REF_PIXEL. An interface may be provided between the memory circuit 122 and the first circuit 124 to present reference pixels in series and/or parallel to the first circuit 124.

The first circuit 124 may be implemented as a
20 two-dimensional array of processing elements (or circuits) 150a-150p, referred to as a processing circuit 150 for short. Each

03-0838
1496.00329

processing element 150a-150p may be configured to calculate an absolute difference between a current pixel from a current block 134 and a respective reference pixel from within a search area. The absolute difference values may be presented in a signal (e.g., ABS_DIFF) In one embodiment, the array may be logically and/or physically arranged as a 16 by 16 matrix of 256 processing elements 150a-150p generating 256 signals ABS_DIFFa-ABS_DIFFp substantially simultaneously. Other arrangements may be implemented for the processing circuit 150 to meet the criteria of a particular application.

The current pixels may be received by the processing circuit 150 along a first edge (e.g., a left edge) of the array via the signals ORIG_PIXEL (e.g., ORIG_PIXELa-ORIG_PIXELk). Once loaded into the first column of processing elements 150a-150p, columns of the current pixels may be shifted through to the other columns of processing elements 150a-150p in the array. Multiple signals (e.g., LEFT_PIXELa-LEFT_PIXELk) may carry a column of the reference pixels from the memory circuit 122 to the processing elements 150a-150p along the first edge of the array. Additional signals (e.g., RIGHT_PIXELa-RIGHT_PIXELk) may carry a row of the reference pixels from the memory circuit 122 to the processing

03-0838
1496.00329

elements 150a-150p along a second edge (e.g., a top edge) of the array. A number of signals (e.g., DOWN_PIXELa-DOWN_PIXELk) may carry another column of the reference pixels from the memory circuit 122 to a third edge (e.g., a right edge) of the array. As
5 with the current pixels, a column/row of reference pixels received along an edge of the array may be shifted through to other columns/rows in the array until each processing element 150a-150p has received a current pixel and a respective reference pixel. By shifting columns and rows of the reference pixels into the
10 processing circuit 124, a current block 134 represented by the loaded current pixels may be moved through the search window of reference pixels, a single position at a time, by loading a new column or a new row of reference pixels, one column or row at a time.

15 The second circuit 126 may be implemented as an adder tree circuit. The adder tree circuit 126 is generally operational to add the absolute difference signals ABS_DIFFa-ABS_DIFFp to generate multiple summation signals (e.g., SUM_BLKa-SUM_BLKd). The adder tree circuit 126 generally comprises multiple adder circuits
20 (or modules) 152a-152d. Each of the adder circuits 152a-152d may

03-0838
1496.00329

be operational to generate one of the sum signals
SUM_BLKa-SUM_BLKd.

The third circuit 128 may be implemented as a mode
decision tree circuit. The mode decision tree circuit 128 is
5 generally operational to add the sum signals SUM_BLKa-SUM_BLKd in
several different combinations to generate a sum of absolute
difference (SAD) value for each of several modes corresponding to
the various block sizes. The mode decision tree circuit 128 may
also be operational to add costs and biases to the SAD values,
10 maintain the minimum or smallest SAD per mode and determine a best
mode (e.g., lowest SAD). The mode decision tree circuit 128
generally presents the signal MODE to indicate the best mode
determined for the current block 134. The motion vector signal MV
may be presented by the mode decision tree circuit 128 with one or
15 more motion vectors corresponding to the best mode. The signal SAD
may be presented by the mode decision tree circuit 128 with a value
for the sum of absolute differences corresponding to the best mode.

Referring to FIG. 3, a block diagram of an example
implementation of the processing element circuit 150a is shown.
20 The processing element 150a generally comprises a multiplexer 160,
a subtraction circuit 162, an absolute value circuit 164 and

03-0838
1496.00329

multiple registers 166a-166d. The multiplexer 160 may receive each of the signals LEFT_PIXELa, RIGHT_PIXELa and DOWN_PIXELa. Depending on the location of the processing element circuit 150a within the array of the processing circuit 124, the signals
5 LEFT_PIXELa, RIGHT_PIXELa and DOWN_PIXELa may be received from an adjacent processing element circuit 150b-150p or the memory circuit 122.

The multiplexer 160 may route the signals LEFT_PIXELa, RIGHT_PIXELa and DOWN_PIXELa to the register 166a, one at a time,
10 based upon the direction the reference pixels are being shifted through the array. The register 166a may present the stored reference pixel to other adjacent (e.g., left, right and down) processing elements in a signal (e.g., REF_PIXELa). The above arrangement generally allows the reference pixels to be shifted
15 through the array left-to-right, right-to-left and top-to-bottom. Other shifting arrangements may be implemented to meet the criteria of a particular application.

The register 166b may receive and store a current pixel from a signal (e.g., ORIG_PIXELa). Depending on the position of
20 the processor element 150 within the array, the signals ORIG_PIXELa may be received from external to the array or from an adjacent

03-0838
1496.00329

processor element 150b-150p. The register 166b may present the stored current pixel to an adjacent processing element 150b-150p (e.g., left) in a signal (e.g., ORIG_OUTa). The above arrangement generally allows the current pixels to be shifted through the array
5 from left-to-right. Other shifting arrangements may be implemented to meet the criteria of a particular application.

The subtractor circuit 162 may be operational to generate a difference value in a signal based on the values of the reference pixel stored in the register 166a and the current pixel stored in
10 the register 166b. Calculation of the difference value may be either the reference pixel value minus the current pixel value or the current pixel value minus the reference pixel value. The difference value may be stored in the optional register 166c.

The absolute value circuit 164 may be operational to
15 generate an absolute difference value from the difference value. The absolute difference value may be stored in the optional register 166d. The register 166d may present the absolute difference value in a signal (e.g., ABS_DIFFa) to the adder tree circuit 126.

20 Referring to FIG. 4, a block diagram of an example implementation of the adder circuit 152a is shown. The adder

03-0838
1496.00329

circuit 152a generally comprises multiple addition circuits 170a-170n and multiple registers 172a-172n. The addition circuits 170a-170n and the registers 172a-172n may be arranged as several stages 174a-174f. The first stage 174a may comprise a portion of
5 the addition circuits 170a-170n and a corresponding number of registers 172a-172n. In general, a single addition circuit 170a-170n may be implemented in the first stage 174a for every two absolute difference signals ABS_DIFFa-ABS_DIFFp received (e.g., an integer fraction of the total). Each successive stage 174b-174f
10 generally has half as many addition circuits 170a-170n and registers 172a-172n as the previous stage 174a-174e. The final stage 174f may have a single addition circuit 170n and a single register 172n. The registers 172a-172n may provide timing control for propagating the absolute difference values through the adder
15 tree circuit 126.

The illustrated adder circuit 152a generally has 64 inputs for receiving a 64-signal portion of the 256 signals ABS_DIFFa-ABS_DIFFm. Therefore, the adder circuit 152a may be operational to generate a sum of absolute difference value for an
20 8Hx8V block of pixels. Therefore, four adder circuits 152a-152d may be implemented to sum all 256 signals ABS_DIFFa-ABS_DIFFp into

03-0838
1496.00329

four sum of absolute difference values expressed in the signals SUM_BLKa-SUM_BLKd. As such, the smallest block, sub-block or partition size supported by the example implementation of the adder circuit 152a may be 8Hx8V. Larger block sizes, such as 8Hx16V,
5 16Hx8V and 16Hx16V may be supported by adding SAD values for the 8Hx8V blocks in the mode decision tree circuit 128. From the four signals SUM_BLKa-SUM_BLKd, the mode decision tree circuit 128 may generate nine SAD values for nine possible partitions of the 16x16 block of pixels.

10 To support a 4Hx4V block, an adder circuit 152a summing
16 of the signals ABS_DIFFa-ABS_DIFFp would generally be sufficient. Therefore, the adder tree circuit 126 may comprise 16 adder circuits 152a-152d and generate 16 sum signals SUM_BLKa-SUM_BLKd. Assuming a 16x16 pixel current block 134, the
15 mode decision tree circuit 128 may combine the 16 signals SUM_BLKa-SUM_BLKd into 41 SAD values in support of all of the block sizes defined by the H.264 specification ("Advanced Video Coding", International Telecommunication Union Telecommunication Standardization Sector, Geneva, Switzerland). Other configurations
20 of the adder circuits 152a-152d and the adder tree circuit 126 may be implemented to meet the criteria of a particular application.

Referring to FIG. 5, a block diagram of an example implementation of the mode decision tree circuit 128 is shown. The mode decision tree circuit 128 generally comprises an add circuit (or module) 180, a storage circuit (or module) 182, a decision circuit (or module) 184, a first bias circuit (or module) 186 and a second bias circuit (or module) 188. The adder circuit 180 may generate multiple intermediate signals (e.g., Ja-Ji), one for each possible partition, based on the signals SUM_BLKa-SUM_BLKd. The storage circuit 182 may examine the signals Ja-Ji and store each minimum SAD value found for the respective partition over the search window and the corresponding motion vectors. The lowest SAD values may be presented to the select circuit 184 via intermediate signals (e.g., Ka-Kd). Each of the motion vectors associated with the respective lower SAD value for a partition may also be presented to the select circuit 184 via intermediate signals (e.g., Ma-Mi).

The select circuit 184 may be operational to determine the best or lowest SAD value among the signals Ka-Kd. The lowest SAD value generally indicates a best match for the partitions of the current block 134 within the search window of the reference frame or frames. The select circuit 184 may present the lowest SAD

03-0838
1496.00329

value found for the current block 134 in the signal SAD. The partition mode associated with the lowest SAD may be presented by the select circuit 184 in the signal MODE. The one or more motion vectors corresponding to the partition mode may also be presented
5 by the select circuit 184 in the signal MV.

The first bias circuit 186 may generate a signal (e.g., COST) conveying a cost (or bias) value. Two signals (e.g., X_DELAYED and Y_DELAYED) may be generated by the first bias circuit 186 indicating the proposed motion vector associated with the cost
10 value. The second bias circuit 188 may generate multiple signals (e.g., SATD_BIASa-SATD_BIASd) for conveying individual bias values for the respective SAD values. The first bias circuit 186 and the second bias circuit 188 may consider a quantization parameter (QP), a magnitude of the motion vector components and the block size in
15 determining the cost and bias values. While variable block size motion estimation generally provides better estimation of small and irregular motion fields than fixed block size motion estimation, supporting different block sizes uses more bits in the bitstream for signaling the size of the blocks and for encoding motion
20 vectors at smaller block sizes. The lookup tables (LUTs) within the first bias circuit 186 and the second bias circuit 188

03-0838
1496.00329

generally permit a rate-distortion tradeoff between block size/numbers motion vectors (e.g., increasing bit rate) and prediction error (e.g., reducing bit rate).

The adder circuit 180 generally comprises multiple adders
5 200a-200n and a circuit 202. The adders 200a-200n may be arranged to sum various combinations of the individual signals SUM_BLKa-SUM_BLKd to calculate an initial SAD value for each possible partition (e.g., MODE0, MODE1_0, MODE1_1, MODE2_0, MODE2_1, MODE3_0, MODE3_1, MODE3_2 and MODE3_3). The circuit 202
10 may be operational to extend the initial SAD values to a predetermined bit length (e.g., 20 bits).

The storage circuit 182 generally comprises multiple modules 210a-210i and a plurality of adders 212a-212i. In generale, a module 210a-210i may be implemented for each partition.
15 Each module 210a-210i generally comprises an adder 214 and a comparator/register 216. The adder 214 may add the cost value to the initial SAD value. The comparator/register 216 is generally operational to compare each SAD value received with a minimum SAD value stored internally. If the newly received SAD value is less
20 than an existing stored SAD value, (i) the new SAD value is stored in place of the existing SAD value and (ii) the corresponding

03-0838
1496.00329

motion vector (e.g., from signals X_DELAYED and Y_DELAYED) is stored in place of an existing motion vector. Therefore the modules 212a-212i may save the best SAD values and the motion vectors found over the whole search area, resulting in an additional cost of only 8 comparator/register elements 216 to support variable block size motion estimation. The adders 212a-212e may be arranged to add the minimum SAD values for the non-mode-zero partitions to generate a single SAD value for the entire current block 134 in each mode (e.g., MODE0, MODE1, MODE2 and MODE3). The adders 212f-212i may be arranged to add the bias values from the signals SATD_BIASa-SATD_BIASd to the respective SAD values to generate the signals Ka-Kd.

The first bias circuit 186 generally comprises a block 220, a block 222, a block 224, a block 226, a lookup table 228, a lookup table 230 and an adder 232. The blocks 220 and 222 may present a quantization parameters for the x and y motions. The block 224 and 226 may present the x and y components of the motion vector currently being assessed. The lookup tables 228 and 230 may generate intermediate values associated with each component of the present motion vector being considered. The adder 232 may add the

03-0838
1496.00329

intermediate values from the LUTs 228 and 230 to generate the cost value in the signal COST.

The second bias circuit 188 generally comprises a block 240, a block 242, a block 244, a block 246, a lookup table 250, a
5 lookup table 252, a lookup table 254 and a lookup table 256. The blocks 240-246 may each be configured to generate a quantization parameter for a particular major mode. Each lookup table 250-256 may convert the quantization parameters into the bias values within the signals SATD_BIASa-SATD_BIASd.

10 The present invention may be implemented in any H.264 encoder with a motion estimation processor. The particular design may be adjusted according to the smallest partition size to be considered and the number of possible partitions available. Also, the present invention may be applied to encoders with motion
15 estimation processors for any other digital video compression standards that allow variable block-size motion estimation (e.g., MPEG-4 Part 2 and H.263+).

The present invention may also be implemented by the preparation of ASICs, FPGAs, or by interconnecting an appropriate
20 network of conventional component circuits (such as conventional circuit implementing a state machine), as described herein,

03-0838
1496.00329

modifications of which will be readily apparent to those skilled in the art(s). As used herein, the term "simultaneously" is meant to describe events that share some common time period but the term is not meant to be limited to events that begin at the same point in
5 time, end at the same point in time, or have the same duration.

While the invention has been particularly shown and described with reference to the preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made without departing from the spirit
10 and scope of the invention.